MULTIMEDIA UNIVERSITY

TABLE NO: | | |

STUDENT ID NO: | | | | | | | | | | |

VENUE: _____

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 2, 2018/2019

TCP1101 – PROGRAMMING FUNDAMENTALS
TCP1231 – COMPUTER PROGRAMMING I

(All sections / Groups)

4th Mar 2019
9.00 am – 11.00 am
(2 Hours)

Q1	
Q2	
Q3	
Q4	
Q5	

Total:

INSTRUCTIONS:

1. This Question paper consists of 13 printed pages (including cover page) with 5 Questions only.

2. Student must attempt **ALL QUESTIONS**. All questions carry equal marks (20 marks). The distribution of the marks for each question is given.

3. Please print all your answers **CLEARLY** in the specific answer boxes provided for each question. If needed you may also write your answers on any blank pages at the end but you must write the corresponding question numbers clearly.

4. Submit this question paper at the end of the examination.

Unless otherwise stated, every code segment provided in every question is contained within a main function, complete with the directive `#include <iostream>` and the statements `using namespace std;` and `return 0;`

QUESTION 1 [20]

(a) Evaluate this. Write your answer onto the box:

```
2+1>3 && (!(7 > 3*2) || (8%3))
```

From (b) to (h), write the output as the result of execution of each code segments:

(b)
```
int x = 3, y = 7;

y = x++  +  --y;

cout << x << " " << y;
```

(c)
```
int x = 10;
do
{
     cout << x << " ";
     x = x - 3;
}while(x > 0);
```

(d)
```
int y = 8;

if(y != 8)
y++;
y++;

cout << y;
```

(e)
```
int m[5] = {7, 9, 3, 1, 5};
cout << *m + 1 << endl;
cout << *(m + 3) << endl;
```

(f)
```
for (int x = 0; x <= 7; ++x)
{
   cout << x << " ";
   if (x / 2 < 3)
      x = x + 2;
}
```

Continued.....

(g)
```
vector<int> v1;

for(int i = 0; i < 8; i++)
     v1.push_back(i);

v1.pop_back();
v1.pop_back();
cout << v1.size();
```

(h)
```
int f1(int num)
{
   int value = 4;
   return (value + num);
   value += 3;
}

int main()
{  cout << f1(10) << endl;
   return 0;
}
```

(i)
```
int f2(int & a)
{
   int b = a + 2;
   a = a * 3;
   cout << a << b << endl;

   return b;
}

int main()
{ int b = 2;
  int a = f2(b);
   cout << a << b << endl;
   return 0;
}
```

(j)
```
int num = 4;
int *p1 = &num;
int *p2 = p1;

*p1 = *p2 + 3;
cout << *p2 << " " << num;
```

Continued.....

QUESTION 2

a) Draw the flowchart for a program that will allow a user to input some positive numbers and then find out which number is the largest one. The program will keep accepting new input numbers until the user enters a negative number where it will stop the input. It will either report the largest number found among all the input numbers, or, if the user only enters a negative number from the start, the program will report "No input." Some sample runs of the program are shown below.

//Sample output 1	//Sample output 2
Please enter a number: 26 Please enter a number: 313 Please enter a number: 4 Please enter a number: -8 The largest numbers is 313	Please enter a number: -5 No input.

[10]

b) Write a complete C++ program for the design described in part (a). [10]

Continued.....

[*This page is left blank for writing the answers.*]

Continued.....

[*This page is left blank for writing the answers.*]

Continued……

QUESTION 3

(a) A teacher wants to create a C++ class (`SimpleStat`) to perform some simple statistics on her class marks. This class will allow her to enter her students' marks, compute the total marks, count the number of marks entered, and store the highest mark.

The code segment below shows the main function that uses the `SimpleStat` class, followed by a sample output:

```
#include <iostream>
using namespace std;

int main()
{
   double mark;
   SimpleStat TC03;

   cout << "Please enter the marks.\n";
   cout << "Enter -1 after the last mark: \n\n";

   cin >> mark;
   while (mark >= 0)
   {
        TC03.addMark(mark);
        cin >> mark;
   }
   cout << "\nYou entered " << TC03.getCount()
        << " marks. \n";
   cout << "The highest mark was " << TC03.getHighest()
        << endl;
   cout << "The average mark was " <<
   TC03.calcAverage() << endl;

   return 0;
}
```

```
// Sample output:
Please enter the marks.
Enter -1 after the last mark:

34.5 43.7 81.4 66.3 92.5 74.8 -1

You entered 6 marks.
The highest mark was 92.5
The average mark was 65.5333
```

Continued.....

(a) Fill up the missing lines in the class definition below: [7]

```
class SimpleStat
{
  private:

   _________________          // The highest mark

   _________________          // The sum of the marks

   _________________          // How many marks entered

   bool isNewHighest(double); // Private class function

  _________________:

   ______________________      // Default constructor
   void addNumber(double);
   double calcAverage();

   double getLargest()
   {
       ________________________
   }

   int getCount()
   {
       ________________________
   }
};
```

(b) Write the C++ implementation of the default constructor of the class `SimpleStat` that will initialize all the data members to zero: [4]

The `calcAverage` method is given below:

Continued.....

```
double SimpleStat::calcAverage()
{
   if (count > 0)
        return sum / count;
   else
        return 0;
}
```

(c) Write the code for the **addMark** method, which will receive a mark. If the mark is a valid entry (positive value) it will add it to the sum, increment the count, and check if it is the highest mark entered so far by using the **isNewHighest** method–if so it will update the highest mark: [5]

(d) Write the code for the **isNewHighest** method which will check whether the mark is the highest mark so far–if so it will return true, or false otherwise: [4]

Continued.....

QUESTION 4

(a) Write a function to swap two integers by using pass-by-reference. [7]

(b) Write a function to swap two integers by using pass-by-address (pointer). [7]

Continued.....

(c) Given the code below, write C++ statements that will count the number of the character 'n' stored in the C-string **word**. Note that you are NOT allowed to use any additional library other than the <iostream>. [6]

```
char word[] = "international";// or any other string
                              // with different size
int countN = 0;
```

Continued.....

QUESTION 5

(a) The following code segment contains an error in how the pointers are handled. Identify and describe the error, then suggest a correction to resolve the error. [4]

```
double *p = new double;
double *s = p;

delete p;
p = NULL;
```

(b) Complete the program below that performs the following tasks: [16]

- Prompts the user to enter the number of row and column for a two-dimensional array.
- Creates a dynamic two-dimensional array of type integer according to the row and column sizes entered by the user above.
- Prompts the user to fill up all the elements into the array.
- Deletes the dynamic two-dimensional array.

```
#include <iostream>
using namespace std;

int main()
{
   int row;
   int col;

   cout << "Enter the number of row and column: ";
   cin >> row >> col;
```

Continued.....

```
// Write your answer in the box:
```

```
    return 0;
}
```

End of paper